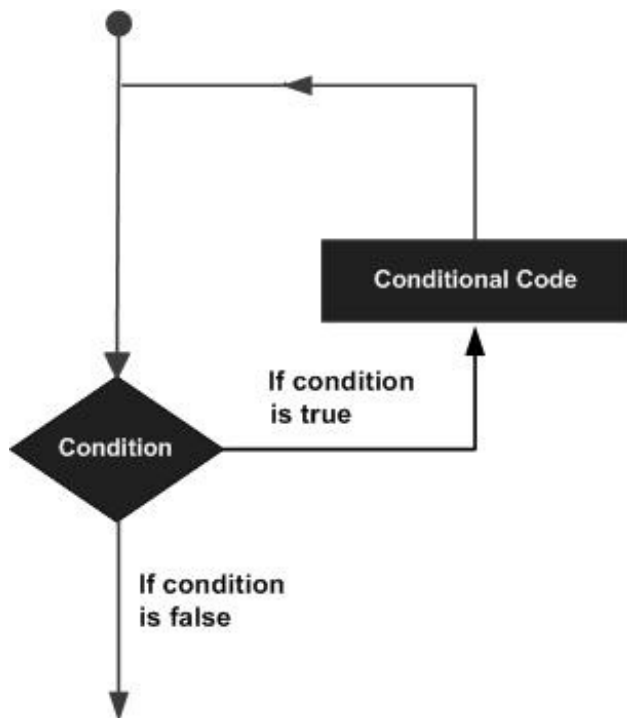


# PYTHON LOOPS

There may be a situation when you need to execute a block of code several number of times. In general statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times and following is the general from of a loop statement in most of the programming languages:



Python programming language provides following types of loop to handle looping requirements. Click the following links to check their detail.

Loop Type	Description
<a href="#">while loop</a>	Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
<a href="#">for loop</a>	Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.
<a href="#">nested loops</a>	You can use one or more loop inside any another while, for or do..while loop.

## Loop Control Statements:

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Python supports the following control statements. Click the following links to check their detail.

Control Statement	Description
<a href="#"><u>break statement</u></a>	Terminates the <b>loop</b> statement and transfers execution to the statement immediately following the loop.
<a href="#"><u>continue statement</u></a>	Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
<a href="#"><u>pass statement</u></a>	The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.